IN THE CLAIMS:

Please amend the claims as follows:

A1

1. (Currently Amended) A method for executing location independent procedure calls in a network system, comprising:

determining a remote node to execute a function, if the function cannot be run on a local node;

executing a route function, wherein the route function comprises:

generating a descriptive data structure for function related data;

generating a pure value buffer; and

flattening the data structure and the pure value buffer into a bundle;

transmitting the ~~function and related~~ flattened data structure and descriptive data to the remote node;

executing ~~the~~ function on the remote node; and

transmitting the results of the function to the local node.

2. (Currently Amended) The method of claim 1, wherein generating a descriptive data structure comprises generating a DTSTRUCT. ~~executing a route function further comprises:~~

~~generating a parameter representative of the function related data; and~~

~~packaging the function related data and the generated parameter for transmission to the remote node~~.

3. (Currently Amended) The method of claim 1 ~~2~~, wherein the pure value buffer comprises a data-only buffer ~~parameter representative of the function related data further comprises a text string, wherein each character in the text string corresponds to a particular data type~~.

4. (Currently Amended) The method of claim 1 ~~2~~, further comprising:

determining if the ~~packaged function related data~~ bundle is cacheable;

determining if the ~~packaged function related data~~ bundle is available in cache memory if it is determined to be cacheable; and

A1

retrieving a cached reply from the cache memory ~~of~~ if the ~~packaged function related data~~ bundle is determined to be cacheable and available in cache memory.

5. (Original) The method of claim 1, wherein determining a remote node to execute a function further comprises reading a parameter associated with the function, wherein the parameter associated with the function indicates where the function may be executed.

6. (Currently Amended) The method of claim 1 ~~2~~, wherein ~~packaging the function related data and the generated parameter further comprises~~ flattening comprises assembling each variable argument indicated in the route function into a buffer.

7. (Currently Amended) The method of claim 1, wherein executing the function on the remote node further comprises:

receiving the bundle ~~function and related data~~ on the remote node;
unpackaging the bundle ~~function related data~~ on the remote node;
computing the function on the remote node; and
packaging a function reply.

8. (Currently Amended) The method of claim 7, wherein packaging a function reply further comprises flattening the function reply.

9. (Currently Amended) The method of claim 7, wherein unpackaging the bundle ~~function related data~~ further comprises unflattening the bundle ~~function related data~~.

10. (Currently Amended) The method of claim 8 ~~1~~, further comprising:

receiving the transmitted results of the function on the local node;
determining if the transmitted results are cacheable; and

A1

storing the transmitted results in a cache memory if the transmitted results are determined to be cacheable.

11. (Original) The method of claim 1, further comprising:

queuing at least one of pre-flattened commands and flattened commands prior to transmission to a remote node; and

cooperatively executing the queued commands in a single network transaction.

12. (Currently Amended) A method for transparently executing function calls from a local node on a remote node, comprising:

determining a remote node for execution of a function;

calling a route function configured to generate a flattened pure value buffer containing function related data;

transmitting the buffer ~~function related data~~ from the local node to the remote node;

executing the function on the remote node; and

transmitting results of the function to the local node.

13. (Original) The method of claim 12, wherein determining a remote node further comprises reading a parameter associated with the function, wherein the parameter indicates the remote node for execution of the function.

14. (Original) The method of claim 12, wherein calling a route function further comprises:

generating a text string, wherein each element of the text string identifies the data type of a portion of the function related data; and

bundling the function related data.

15. (Original) The method of claim 14, wherein the text string further comprises a DTSTRUCT string.

A1

16. (Original) The method of claim 14, wherein bundling further comprises flattening the function related data.

17. (Currently Amended) The method of claim 12, wherein executing the function on the remote node further comprises:

unbundling the function related data;
computing the function; and
bundling ~~the~~ reply to the function.

18. (Original) The method of claim 17, wherein unbundling and bundling further comprise unflattening and flattening, respectively.

19. (Original) The method of claim 17, further comprising looking up a function pointer that indicates the location of the function to the remote node.

20. (Original) The method of claim 12, further comprising determining if the function related data is cacheable and storing the function related data in cache memory if the function related data is determined to be cacheable.

21. (Currently Amended) The method of claim 12, further comprising determining if the results of the function are cacheable and storing the results of the function in cache memory if ~~is~~ the results of the function are determined to be cacheable.

22. (Currently Amended) The method of claim 12, wherein transmitting the results of the function to the local ~~remote~~ node further comprises unflattening the results.

23. (Currently Amended) The method of claim 12, further comprising:

queuing at least one of pre-flattened commands and flattened commands prior to transmission to the ~~a~~ remote node; and

cooperatively executing the queued commands in a single network transaction.

A1

24. (Currently Amended) A computer readable medium storing a software program that, when executed by a processor, causes the processor to perform a method comprising:

determining a remote node to execute a function, if the function cannot be run on a local node;

executing a route function configured to assemble a flattened pure value buffer containing function related data;

transmitting the buffer ~~function and related data~~ to the remote node;

executing the function on the remote node; and

transmitting the results of the function to the local node.

25. (Original) The computer readable medium of claim 24, wherein executing a route function further comprises:

generating a parameter representative of the function related data; and

packaging the function related data and the generated parameter for transmission to the remote node.

26. (Original) The computer readable medium of claim 25, wherein the parameter representative of the function related data further comprises a text string, wherein each character in the text string corresponds to a particular data type.

27. (Original) The computer readable medium of claim 25, further comprising:

determining if the packaged function related data is cacheable;

determining if the packaged function related data is available in cache memory if it is determined to be cacheable; and

retrieving a cached reply from the cache memory of the packaged function related data is determined to be cacheable and available in cache memory.

28. (Original) The computer readable medium of claim 24, wherein determining a remote node to execute a function further comprises reading a parameter associated

with the function, wherein the parameter associated with the function indicates where the function may be executed.

A1

29. (Original) The computer readable medium of claim 25, wherein packaging the function related data and the generated parameter further comprises flattening each variable argument indicated in the route function into a buffer.

30. (Original) The computer readable medium of claim 24, wherein executing the function on the remote node further comprises:

receiving the function and related data on the remote node;
unpackaging the function related data on the remote node;
computing the function on the remote node; and
packaging a function reply.

31. (Original) The computer readable medium of claim 30, wherein packaging a function reply further comprises flattening the reply.

32. (Original) The computer readable medium of claim 30, wherein unpackaging the function related data further comprises unflattening the function related data.

33. (Original) The computer readable medium of claim 24, further comprising:

receiving the transmitted results of the function on the local node;
determining if the transmitted results are cacheable; and
storing the transmitted results in a cache memory if the transmitted results are determined to be cacheable.

34. (Original) The computer readable medium of claim 24, further comprising:

queuing at least one of pre-flattened commands and flattened commands prior to transmission to a remote node; and

cooperatively executing the queued commands in a single network transaction.

A\

35. (Currently Amended) A computer readable medium storing a software program that, when executed by a processor, causes the processor to perform a method comprising:

determining a remote node for execution of a function;

calling a route function;

generating a flattened pure value buffer containing function related data and a descriptive data structure;

transmitting the buffer ~~function related data~~ from the local node to the remote node;

executing the function on the remote node; and

transmitting results of the function to the local node.

36. (Original) The computer readable medium of claim 35, wherein determining a remote node further comprises reading a parameter associated with the function, wherein the parameter indicates the remote node for execution of the function.

37. (Original) The computer readable medium of claim 35, wherein calling a route function further comprises:

generating a text string, wherein each element of the text string identifies the data type of a portion of the function related data; and

bundling the function related data.

38. (Original) The computer readable medium of claim 37, wherein the text string further comprises a DTSTRUCT string.

39. (Original) The computer readable medium of claim 37, wherein bundling further comprises flattening the function related data.

40. (Original) The computer readable medium of claim 35, wherein executing the function on the remote node further comprises:

unbundling the function related data;

computing the function; and

A1

bundling the reply to the function.

41. (Original) The computer readable medium of claim 40, wherein unbundling and bundling further comprise unflattening and flattening, respectively.

42. (Original) The computer readable medium of claim 40, further comprising looking up a function pointer that indicates the location of the function to the remote node.

43. (Original) The computer readable medium of claim 35, further comprising determining if the function related data is cacheable and storing the function related data in cache memory if the function related data is determined to be cacheable.

44. (Original) The computer readable medium of claim 35, further comprising determining if the results of the function are cacheable and storing the results of the function in cache memory is the results of the function are determined to be cacheable.

45. (Original) The computer readable medium of claim 35, wherein transmitting the results of the function to the remote node further comprises unflattening the results.

46. (Original) The computer readable medium of claim 35, comprising:

queuing at least one of pre-flattened commands and flattened commands prior to transmission to a remote node; and

cooperatively executing the queued commands in a single network transaction.